UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/663,563	09/15/2000	Thomas E. Saulpaugh	5181-72000	3700

58467   7590   02/21/2008

MHKKG/SUN
P.O. BOX 398
AUSTIN, TX 78767

EXAMINER
KANG, INSUN

ART UNIT	PAPER NUMBER
2193	

MAIL DATE	DELIVERY MODE
02/21/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

PTOL-90A (Rev. 04/07)

	Application No.	Applicant(s)
Office Action Summary	09/663,563	SAULPAUGH ET AL.
	Examiner	**Art Unit**	
	INSUN KANG	2193	

-- *The MAILING DATE of this communication appears on the cover sheet with the correspondence address* --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE _3_ MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
  Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on _30 November 2007 and 14 February 2008_.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) _1, 2, 4-41, 43-71, and 73-90_ is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☒ Claim(s) _1,2,4-11,40,41,43-49,71 and 73-77_ is/are allowed.

6)☒ Claim(s) _12-14, 20-39, 50-52, 54-70, 78, 81-90_ is/are rejected.

7)☒ Claim(s) _15-19,53,79 and 80_ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/ are: a)☐ accepted or b)☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a)☐ All b)☐ Some * c)☐ None of:

      1.☐ Certified copies of the priority documents have been received.

      2.☐ Certified copies of the priority documents have been received in Application No. _____.

      3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
    Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

1.   This action is in response to the amendment filed on 11/30/2007 and 2/14/2008.

2.   Claims 1, 2, 4-41, 43-71, and 73-90 are pending in the application.

Allowable Subject Matter

3.   Claims 1, 2, 4-11, 40, 41, 43-49, 71, 73-77 are allowed. Claims 15-19, 53, and 79-80 are

objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in

independent form including all of the limitations of the base claim and any intervening claims.

The following is an examiner's statement of reasons for allowance: Claims 1, 2, 4-11, 40,

41, 43-49, 71, 73-77 are allowed. Johnson and Allen disclose conversion from/to XML.

However, Johnson and Allen do not disclose conversion using an intermediary table

representation of an object.

Any comments considered necessary by applicant must be submitted no later than the

payment of the issue fee and, to avoid processing delays, should preferably accompany the issue

fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for

Allowance."

Terminal Disclaimer

4.   The terminal disclaimer filed on 2/14/2008 has been recorded.

Claim Rejections - 35 USC § 103

5.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or
> described as set forth in section 102 of this title, if the differences between the subject

matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6.   Claims 12-14, 20-39, 50-52, 54-70, 78, 81-90 are rejected under 35 U.S.C. 103(a) as being unpatentable over Johnson ("XML JavaBeans Integration, Part 3," 7/1999) in view of Allen (US Patent 6,658,625).

Per claim 12:

Johnson teaches:

-a virtual machine receiving a data representation language representation of a first computer programming language object from a first process a decompilation process of the virtual machine generating the first object from the data representation language representation of the first object, wherein the first object is an instance of a class in the computer programming language (i.e. XMLBeans can transform a JavaBean in memory into an XML document," page 1 first paragraph, line 2; "read and write JavaBeans objects as XML documents," page 1 paragraph 7, line 3; can transform an XML document...into a running JavaBean," page 1 first paragraph, lines 1-3).

Johnson does not explicitly teach and the decompilation process of the virtual machine providing the first object to a second process executing within the virtual machine. However, Allen teaches it was known in the pertinent art, at the time applicant's invention was made, to reconstitute the object state in a distributed system (i.e. col. 3 lines 12-24). It would have been obvious for one having ordinary skill in the art to modify Johnson's disclosed system to

incorporate the teachings of Allen. The modification would be obvious because one having

ordinary skill in the art would be motivated to reproduce object state in a distributed system.

Per claim 13:

The rejection of claim 12 is incorporated, and further, Johnson teaches:

- the first object references one or more computer programming language objects,

wherein the representation of the first object includes representations of the one or more

referenced objects (i.e. page 5 paragraph 3-4).

Per claim 14:

The rejection of claim 13 is incorporated, and further, Johnson teaches:

- the decompilation process generating the one or more referenced objects from the

representations of the one or more referenced objects included in the representation of the first

object (i.e. page 5 paragraph 5).

Per claim 20:

The rejection of claim 12 is incorporated, and further, Johnson teaches:

-the data representation language representation of the first object comprises an identifier

of the class of the first object, and wherein the decompilation process generating the first object

from the data representation language representation of the first object comprises instantiating

the first object as an instance of the class associated with the class identifier (i.e. page 5

paragraph 3-5).

Per claim 21:

The rejection of claim 16 is incorporated, and further, Johnson teaches:

-providing an application programming interface (API) for the decompilation process, wherein the API comprises interfaces to one or more methods of the decompilation process configured for use by processes executing within the virtual machine to generate computer programming language objects from data representation language representations of the objects. (i.e. page 7 last paragraph, lines 1-4).

Per claim 22:

Johnson teaches:

-said data representation language is extensible Markup Language (XML) (i.e. "XML document," page 1, lines 2-3).

Per claim 23:

Johnson teaches:

- said computer programming language is the Java programming language (i.e. "a JavaBean," page 1, lines 2-3).

Per claim 24:

Johnson teaches:

-the virtual machine is a Java Virtual Machine (JVM) (i.e. "running JavaBean," page 1, line 3).

Per claim 25:

Johnson teaches:

-a first virtual machine receiving from a first process a computer programming language object, wherein the object is an instance of a class in the computer programming language the first virtual machine generating a representation of the object in a data representation language subsequent to said receiving generating the data representation language representation of the object (i.e. XMLBeans can transform a JavaBean in memory into an XML document," page 1 first paragraph; "read and write JavaBeans objects as XML documents," page 1 paragraph 7; " can transform an XML document…into a running JavaBean," page 1 first paragraph).

Johnson does not explicitly teach generating a message in the data representation language and sending the message to a second process. However, Allen teaches generating XML message of an object and providing such XML representation to another process was known in the pertinent art, at the time applicant's invention was made, to reconstitute the object state from the decompilation in a distributed system (i.e. col. 3 lines 12-24). It would have been obvious for one having ordinary skill in the art to modify Johnson's disclosed system to incorporate the teachings of Allen. The modification would be obvious because one having ordinary skill in the art would be motivated to reproduce object state in a distributed system.

Johnson in view of Allen further teaches: the second process generating a copy of the computer programming language object from the data representation language representation of the object included in the message (i.e. page 1 first paragraph).

Per claim 26:

The rejection of claim 25 is incorporated, and further, Johnson teaches:

-the object references one or more computer programming language objects, and wherein

said generating a representation of the object in a data representation language comprises

generating data representation language representations of the one or more objects(i.e. page 5

paragraph 3-5).

Per claim 27:

The rejection of claim 25 is incorporated, and further, Johnson teaches:

-for each of the one or more instance variables in the object, generating an element

in the data representation language representation of the first object, wherein the element for

each of the one or more instance variables includes an identifier of the instance variable and a

value of the instance variable (i.e. page 7 last paragraph, lines 1-4).

Per claim 28:

The rejection of claim 25 is incorporated, and further, Johnson teaches:

-the second process receiving the message including the data representation

language representation of the object; the second process providing the data representation

language representation of the object to a second virtual machine; the second virtual machine

generating the copy of the object from the data representation language representation of the

object; and the second virtual machine providing the copy of the object to the second process

(i.e. page 5 paragraph 3-4).

Per claim 29:

The rejection of claim 28 is incorporated, and further, Johnson teaches:

-the first object references one or more computer programming language objects, wherein the data representation language representation of the first object includes data representation language representations of the one or more referenced objects, and wherein said generating the copy of the object from the data representation language representation of the object comprises generating copies of the one or more referenced objects from the data representation language representations of the one or more referenced objects (i.e. page 5 paragraph 5).

Per claim 30:

The rejection of claim 28 is incorporated, and further, Johnson teaches:

-instantiating the copy of the object as an instance of the class; and

for each of the one or more elements in the data representation language

representation of the object, initializing a corresponding instance variable

in the copy of the object in accordance with the element (i.e. page 7 last paragraph lines 1-4).

Per claims 31-33, they are another method versions of claims 22-24, respectively, and are rejected for the same reasons set forth in connection with the rejection of claims 22-24 above.

Per claim 34-39: they are another method versions of claims 25-33, respectively, and are rejected for the same reasons set forth in connection with the rejection of claims 25-33 above.

Per claim 50:

Johnson discloses:

-receive a data representation language representation of a first computer programming language object (i.e. XMLBeans can transform a JavaBean in memory into an XML document," page 1 first paragraph)

- generate the first object from the data representation language representation of a first object, wherein the first object is an instance of a class in the computer programming language (i.e. "read and write JavaBeans objects as XML documents," page 1 paragraph 7; "can transform an XML document...into a running JavaBean," page 1 first paragraph).

Johnson does not explicitly teach a system comprising a processor and memory to execute the instructions. Allen teaches it was known in the pertinent art, at the time applicant's invention was made, to reconstitute the object state in a distributed system (i.e. col. 3 lines 12-24). It would have been obvious for one having ordinary skill in the art to modify Johnson's disclosed system to incorporate the teachings of Allen. The modification would be obvious because one having ordinary skill in the art would be motivated to execute and reproduce object state in a distributed system.

Per claims 51-52 and 54-61, they are another device versions of claims 12-14 and 20-24 respectively, and are rejected for the same reasons set forth in connection with the rejection of claims 12-14 and 20-24 above.

Per claims 62-70, they are the medium versions of claims 25-33, respectively, and are rejected for the same reasons set forth in connection with the rejection of claims 25-33 above.

Per claims 78 and 81-83, they are the medium versions of claims 12-14 and 20-24,

respectively, and are rejected for the same reasons set forth in connection with the rejection

of claims 12-14 and 20-24 above.

Per claims 84-90, they are the medium versions of claims 25-33, respectively, and are

rejected for the same reasons set forth in connection with the rejection of claims 25-33 above.

Response to Arguments

7.   Applicant's arguments filed on 11/30/2007 have been fully considered but they are not

persuasive.

The applicant states that per claim 12, applicant's claim does not recite merely

reconstituting the object state in a distributed system. Allen does not teach a decompilation

process generating the object from the data representation language representation of the object

and providing the object to a second process executing within the virtual machine. Nowhere

does either Johnson or Allen mention providing a data representation language representation of

an object to a second process. Allen fails to mention sending a data representation language

representation of an object to a second (different) process (remark, 4-5).

In response, the claim does not recite that the second process being a "different process

than the first process from which the data representation language representation of a first

computer programming language object was received." Furthermore, Allen discloses a

conversion from XML to an object (i.e. the XML data into a hash table which is an object, col.

12 lines 10-20). This conversion is the decompilation process of XML into an object. Allen also discloses providing such XML representation to another process in a distributing system, to reconstitute the object state from the decompilation in a distributed system (i.e. col. 3 lines 12-24). In addition, col. 7, lines 12-27 detail that the invention changes data from one format to another format between applications, wherein the system itself can be a single processor, client computer system or single user device (col. 7, lines 27-50). Therefore, the applications run on the same device and communicate with each other by translating the data / messages between different formats.

The applicant states that: per claim 25, Allen uses XML to describe data formats not for messages including data representation language representations of objects. Allen's system does not involve sending messages in a data representation language. Allen is not concerned with nor teach anything regarding reproducing object state in a distributed system. The examiner fails to actually provide any reason as to why one would include the generic data conversion system of Allen into the XMLJabaBean system of Johnson (remark, 6-7).

Allen also discloses a data conversion between formats and types (col. 1 lines 5-10) in a distributed system with API layer that provides interfaces for clients and services from a server (col. 2 lines 23-26; col. 7 lines 50-61). The interfaces are for messages communicating between objects. In Allen, the data conversion from/to XML is used to send and/or received data from a server (col. 6 lines 25-34) in XML. The conversion from the XML message reproduce the data presented in the XML format. As stated above, Johnson does not explicitly teach generating a message in the data representation language and sending the message to a second process. However, Allen teaches generating XML message of an object and providing such XML

representation to another process in a distributing system, to reconstitute the object state from the

decompilation in a distributed system (i.e. col. 3 lines 12-24).

The applicant states that: per claim 34, Allen's data descriptions are simply not messages

in a data representation language. Instead, Allen's data descriptions are stored in and XML file

that is read and parsed not send as a message by Allen's system (remark, 8).

In response, in Allen's distributed system where API layer provides interfaces for clients

and services, the interfaces are for message communicating between objects. The XML data are

send/received as a message in the distributed system (col. 2 lines 23-26; col. 7 lines 50-61).

Conclusion

8.   **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time

policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE

MONTHS from the mailing date of this action. In the event a first reply is filed within TWO

MONTHS of the mailing date of this final action and the advisory action is not mailed until after

the end of the THREE-MONTH shortened statutory period, then the shortened statutory period

will expire on the date the advisory action is mailed, and any extension fee pursuant to 37

CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event,

however, will the statutory period for reply expire later than SIX MONTHS from the mailing

date of this final action.

9.   Any inquiry concerning this communication or earlier communications from the

examiner should be directed to INSUN KANG whose telephone number is (571)272-3724. The

examiner can normally be reached on M-F 8:30-5 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, Lewis A. Bullock can be reached on 571-272-3759. The fax phone number for the

organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent

Application Information Retrieval (PAIR) system. Status information for published applications

may be obtained from either Private PAIR or Public PAIR. Status information for unpublished

applications is available through Private PAIR only. For more information about the PAIR

system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR

system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would

like assistance from a USPTO Customer Service Representative or access to the automated

information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Insun Kang/
Examiner, Art Unit 2193

/Lewis A. Bullock, Jr./
Supervisory Patent Examiner, Art Unit 2193